

**METHOD AND APPARATUS FOR A LOOSELY COUPLED, SCALABLE  
DISTRIBUTED MULTIMEDIA STREAMING SYSTEM**

5           FIELD OF THE INVENTION

          This invention relates generally to the field of distributed multimedia streaming and more particularly to media content distribution for high bit rate streaming from distributed components

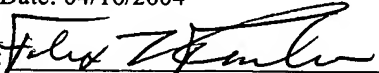
10           BACKGROUND OF THE INVENTION

          High bit rate multimedia streaming, particularly high bit rate video streaming has evolved from handling thousands of simultaneous subscriber to millions of subscribers. The conventional system architecture based on a single powerful machine or a cluster system with central control can no longer meet the massive demands.

15           SUMMARY OF THE INVENTION

          A scalable distributed multimedia streaming system employs at least one media station having a media director and a plurality of media engines. Each media engine incorporates media content storage, communications channels for retrieving media  
20   content over the network and communications channels for streaming media content over the network. The media director has a controller adapted for directing retrieval over the network of media content by a selected media engine, tracking content stored on the

Certification under 37 C.F.R. §1.10  
This correspondence is being filed by Express mail addressed to  
Commissioner for Patents, P.O. Box 1450  
Alexandria, VA 22313-1450  
Express Mail No.: ED 044188786  
on Date: 04/16/2004

By:   
Felix L. Fischer, Registration No. 31,614

media engines and redirecting a content requested from a media console connected to the media station to a selected one of the media engines storing content corresponding to the request for streaming. Multiple media stations are employed to expand the network using a media location registry communicating with the media director in each media station.

- 5 The media location registry is a central repository for storing the location of all media content in the media stations. Downloaded content can then be presented by the media stations to the media consoles connected to them through a network and intercommunication between the media stations for transfer of content can also be accomplished through the network.

10

#### BRIEF DESCRIPTION OF THE DRAWINGS

These and other features and advantages of the present invention will be better understood by reference to the following detailed description when considered in connection with the accompanying drawings wherein:

- 15 FIG. 1 is a block diagram of the elements incorporated in a media station;

FIG. 2 is a block diagram of a network system employing media stations according to the present invention;

FIG. 3a is a diagram of the hardware interaction and process for streaming data to a subscriber's media console;

- 20 FIG. 3b is a flow diagram of the process for streaming data as shown in FIG. 5a;

FIG. 4 is a flow diagram of the process for rapid replication of segments on alternative media engines to relieve overload;

FIG. 5 is a flow diagram of the process for media engine swapping for avoiding errors in response to subscriber commands;

- 25 FIG. 6 HMFS;

FIG. 7 is a top level block diagram of the hardware physical structure;

FIG. 8 is a detailed block diagram of the chassis arrangement;

FIG. 9 is a block diagram of the functional interaction of the blade main board with the Network Management System and the chassis blade controller;

FIG. 10 is a block diagram of the basic elements of the secret key system for access control in a system employing the invention; and

5        FIG. 11 is a block diagram of the system communication for authentication of a media console request for streaming data.

## DETAILED DESCRIPTION OF THE INVENTION

10        A media content distribution system incorporating the present invention employs a self-sufficient streaming unit designated a media station covering a set of subscribers. Media stations in a typical application are installed in a CO of a broadband network to which the subscribers are connected. The placement of media stations is determined according to the number of subscribers to be covered, network topology and available bandwidth of the network.

15        As shown in FIG. 1 for an exemplary embodiment, each media station 102 incorporates a media director 104 having an EPG server 106 and an application server 108 for handling streaming and trick-mode requests from the subscriber. A Hyper Media File System (HMFS) 110 is incorporated for data storage. A standby media director 104S with identical capabilities is provided to assume the role of the active director upon  
20        failure or removal from service. Multiple media engines 112 are present in the media station. The media director records the location of all programs in the system and which media engine holds a particular program or portions of it. Upon communication from a subscriber media console, the media director directs the media console to the appropriate media engine to begin the data stream. A distributed storage subsystem (for the  
25        embodiment shown, a HMFS) 114 is present in each media engine to employ a large number of independent, parallel, I/O channels 116 to meet massive storage size demands and I/O data rate demands. Media engines are connected together through a set of Gigabit Ethernet switch 118, and to the network 120 communicating with the subscribers.

Matching bandwidth between the network to subscribers and I/O channels avoids any bottleneck in the streaming system.

Each media program (a movie, a documentary, a TV program, a music clip, etc.) is partitioned into smaller segments. Such partitioning provides a small granularity for media data units and makes data movement, replications, staging and management much easier and more efficient.

FIG. 2 demonstrates one embodiment of a network system designated a Media Switch with incorporates groups of media stations configured for use in a number of geographical areas or cities 202 served. A complete description of the Media Switch is disclosed in companion application Attorney Docket No. U001 100084 entitled METHOD AND APPARATUS FOR MEDIA CONTENT DISTRIBUTION IN A DISTRIBUTED MULTIMEDIA STREAMING SYSTEM having a common assignee with the current application, the contents of which are fully incorporated herein by reference. The scalability of the system employing the present invention is demonstrated in FIG. 2. Each city employs a series of media stations 102 interconnected through the metropolitan area network (MAN) 204. Each media station serves a number of subscribers having media consoles 206. Each subscriber has a primary media station to serve its streaming requests. Additionally, each city incorporates on-line support layer elements including a media location registry (MLR) 208, a home media station 210 and a content manager 212 in a data center (DC) 214. For the embodiment shown, a principal city 202' is chosen as a headquarters site. Associated with that site is a Media Asset Management (MAM) system 124. In alternative embodiments, multiple cities incorporate a MAM for introduction of content into the system.

The MAM determines when and where to distribute a program. The CM publishes the program at the time specified by the MAM and the MLR identifies the location of the data for distribution

For streaming content to subscribers, the media director in each of the media stations employs a load balancing scheme to keep track of the task load of the media

engines in the media station. Load balance is achieved by directing streaming requests according to current system states and load distribution. An example of the communications sequence for data transfer under the command of the media director is shown in FIG. 3a with representative IP address locations for the system elements. The

5 media console 206 requests 302 a segment 0021 from the media director 104. The media director identifies the location of the segment in a segment location table 304 as present in media engines 1 and 8, (ME1 and ME8) and redirects 306 the MC to ME1's IP address 10.01.1.11. The MC then requests 308 segment 0021 from ME 1 which begins streaming data 310. When the segment being streamed nears its end, ME1 requests 312 the location

10 of the next segment from the MD which locates the next segment and MEs storing that segment in the segment location table, selects an ME based on load and status and replies 314 with the identification of the next segment (seg 0022) and the IP address 10.0.1.12 of ME2 where the next segment resides. ME1 notifies ME2 to preload 316 the next segment seg 0022 and upon completion of the streaming of seg 0021 directs 318 ME2 to start

15 streaming seg 0022 to IP address 18.0.2.15, the media console. ME2 then begins streaming 320 the data from seg 0022 to the MC.

A flow diagram of the sequence described with respect to FIG. 3a is shown in FIG. 3b. Upon assumption of the communication of the stream with the MC by ME2, ME2 sends a notification 322 to the MD. The process described continues until the MC

20 orders a cessation of streaming 324 by the ME at which time the ME notifies the MD the streaming has stopped 326. The media director may employ a number of MEs to supply the segments in sequence to the media console. Flexibility in assignment of ME based on content and load allows the MD to balance the operation of the MEs.

As a portion of the load balancing scheme, a rapid replication scheme is used to

25 copy a segment from one media engine to another. When a media engine exceeds its capacity of streaming, a highly demanded segment can be replicated to another media engine and further requests for that segment are directed to the new media engine. The

extra delay observed by the streaming request that triggered the replication is less than 30 milliseconds in exemplary embodiments.

The communications sequence is shown in FIG. 4. A first media console MC1 requests streaming 402 of a segment to the Media director MD. The MD replies 404 with a redirection to a media engine ME1 storing the segment. MC1 requests playing of the stream 406 from ME1 and ME1 responds 408 by streaming the RTP packets of data from the segment. The MD has cataloged the redirection to ME1 and monitors ME1's load. If ME1 has reached a predetermined maximum threshold (some percentage of the maximum capacity), when another media console MCn requests streaming 410 of the same segment, if the segment is not present on another available ME in the segment location table, the MD directs 412 another media engine ME2 to fetch the segment and specifies the ME from which the segment is to be replicated. In various embodiments the maximum threshold may be determined such that the replication can occur from the first media engine or other existing media engines in the segment location table. Alternatively, the fetch command may direct copying of the segment from a media engine in another media station as described with respect to FIG. 7. For purposes of the example, the source media engine defined by the MD is designated MEx. ME2 requests a copy 414 of the segment from MEx which replies by sending the segment 416. Upon direction of the fetch, the MD replies 418 to MCn redirecting to the IP address of ME2. MCn then requests playing of the stream 420 and ME2 responds 422 forwarding RTP packets for the segment to MCn. When copying of the segment from MEx to ME2 is complete, ME2 sends a copy done 424 to the MD which notifies the MLR of the new location for the segment as previously described.

A stream swapping method is used to exchange two streams of the same segment, one on a first media engine ME2 that has a complete copy of the segment and a second on a second media engine ME1 which is currently receiving the same segment. Where the subscriber attempts a fast-forward while streaming from ME1 with the incomplete segment, the media director swaps the fast-forwarding stream from ME1 to ME2 (with

the complete segment). The stream using the same segment running at normal rate is swapped from the first media engine to the second media engine thereby avoiding a failure of the fast forwarding operation.

FIG. 5 demonstrates the communications sequence for swapping media engines.

5 During normal operation, the media director MD has directed ME1 to fetch 502 a particular segment. ME1 requests a copy 504 of the segment from the source ME (arbitrarily identified as MEx) and MEx responds by sending 506 the desired segment. During receipt of the segment, a media console MC1 requests a stream 508 from the MD which replies 510 redirecting the MC to ME1. MC1 requests playing of the stream 512 and ME1 responds 514 by sending the RTP packets from the requested segment. If MC1  
10 requests a fast forward 516 of the stream (segment) ME1 identifies the potential for a streaming error if the fast forward exceeds the portion of the segment which has been received from MEx. ME1 notifies 518 the MD of the impending error state and the MD replies with the identification of a media engine ME2 (which can be MEx itself) having  
15 the entire segment that is idle or has started streaming after ME1. ME2 has been streaming RTP packets 520 of the segment to another media console MCn. ME1 requests a swap 522 identifying MC1 as the media console in current communication and providing the segment number and frame within the segment. ME 2 begins streaming of data 524 from the segment to MC and, if ME2 has been streaming, returns a swap 526  
20 identifying media console MCn and the frame of the segment. ME1 takes over streaming of RTP packets 528 to MCn.

The media engines in the media station are symmetrical with respect to input and output thereby allowing data to be taken into the media engine substantially as rapidly as streaming data is sent out. As shown in FIG. 6, each media engine employs an HMFS  
25 have multiple storage drives 602. A content program, e.g. a movie, is divided into a sequence of segments. Each segment represents several minutes of contents, 4 minutes for example. In each media station, a segment is stored in at least two media engines, for fault tolerance. The media director in each media station has the database containing the

locations of each segment held by that media station, which is the top level directory of HMFS. For each segment, the directory entry contains the information such as, data size, frame count, frame index, key frame (or I frame) index, inter-frame time interval, media type (MPEG2, MPEG4, WM9, H.264, etc.), time of recording, pointers to disk blocks  
5 holding the data of the segment.

Data in a segment is partitioned into “datalet”, which is the minimum disk I/O unit and buffer allocation unit. For each outgoing stream (stream that is being sent to an MC), a number of buffers 604 connected to a bus 606 from the drive units are used to pre-fetch datalets for the stream. Datalets are distributed to the disk drives in a media  
10 engine (for the embodiment shown in FIG. 6 four drives), so when large number of streams are active on the media engine, all four drives, and associated I/O channels are working in parallel to achieve maximum possible I/O throughput through the buffers to the Gigabit Ethernet switches.

The I/O operations on each disk are optimized by performing the operations in the sequence of their disk addresses so the seek time is minimized. A disk controller 608  
15 operating in concert with an I/O controller module 610 provides sequencing control.

The network interfaces of the media engines are full-duplex Gigabit Ethernet, which provide up to gigabit/second bandwidth in either direction, incoming into a media engine or output from a media engine. The incoming data is buffered in the same fashion  
20 as the output data, and the incoming data is written to the disk in the same pattern as the data is read from disk.

Therefore, the media station can be used as a high bit rate, massive storage repository. This architecture is specifically beneficial in live broadcast transmission where the program segments are transferred to the media stations in real time and  
25 streamed to the media consoles.

For content which is not yet present, or not complete, on the media stations but available on the system, a request from a subscriber results in transfer of the content as shown in FIG. 7. The subscriber media console 206 makes a streaming request 702 to the



media director MS2 MD of the media station MS2. The MD asks 704 the MLR for the location of the program or segment requested. The MLR responds with a notification 706 of locations for the segment. Multiple locations may exist where the desired segment is stored. The MD calculates the relative cost of obtaining the desired copy of the segment  
5 based on a number of parameters including the bandwidth available, distance from the source media station, copying time and load of the source media station. Upon selection of a source media station, MS1 for the example herein, the MD requests 708 the location of the segment from MS1MD which responds 710 with the address of a media engine MS1ME storing the segment. MS2MD then directs 712 a selected media engine MS2ME  
10 to fetch the segment. MS2ME requests 714 a copy of the segment from source media engine MS1ME which responds 716 sending the segment. Upon completion of the copying of the segment, MS2ME notifies 718 the MD of completion of the copy and the MD notifies 720 the MLR of the new location of the segment.

From a hardware standpoint in a representative embodiment, the Media Station  
15 comprises one or more chassis each having multiple individual blades as shown in FIG. 8. The Media Station (MS) 102, a self-contained streaming unit typically located in a CO and covering the vicinity of the CO. Each MS consists of a number of chassis 802. The chassis management system provides external control for the blades in the chassis. Contained within each chassis are blades 804 is the lowest level management unit. Each  
20 blade is an independent computer. It can be either a Media Engine (ME) or a Media Director (MD).

In the embodiment shown, the Media Station is a level of abstraction, with its state represented by its MD. Therefore, the MS need not be an entity in the management structure of a network management system (NMS) 806 employed for hardware control.

25 Network management is a first level of management for the media station(s) and provides a full set of management functionalities and GUI. System load and other operational parameters such as temperature and fan speed are monitored. Automatic alarms can be configured to send email or call to the system operator.

Chassis management is the second level and provides blade presence detection, automatic blade power up, remote blade power up and power down, managed blade power up to avoid current surge during disk drive spin up, chassis id reading and chassis control fail-over.

5 Blade self-management and monitoring is the third level and allows temperature, fan speed, and power supply voltage monitoring and alarm through SNMP to the NMS, self-health monitoring including critical threads monitoring, storage level monitoring, load monitoring, etc. All alarm thresholds can be set remotely by NMS. For software related failures, software restart or OS reboot will be attempted automatically, and the  
10 event will be reported to NMS.

As shown in FIG. 9 for the exemplary embodiment, a chassis can host up to 10 blades 804, each can be a Media Engine or a Media Director. Each blade can read the chassis ID 902 and its own slot number 904 for identification.

All blades in a chassis are equipped with a control unit or Chassis Blade  
15 Controller (CBC) 906. For the exemplary embodiment, each CBC consists of an Intel 8501 chip implementing the control logic and an FPGA configured to act as the control target. The 8501 chip also communicates with the main board 908 through a UART interface 910. The main board can issue control commands or relay control commands received from NMS through the network to the CBC.

20 For the exemplary embodiment, blades located in slot 5 and 6 are the control blades. One active and one standby determined by the arbitration logic at power up. When the chassis is being powered up, the blades in slots 5 and slot 6 arbitrate and one becomes the active controller or media director. The CBC on the active control blade scans the back-plane and powers up the blades in a controlled sequence with a pre-  
25 determined interval to avoid current surge caused by disk drive spin up on the individual blades.

The CBC on the active control blade then scans all slots on the backplane and detects the presence and status of each blade. The standby control blade monitors the

status of the active control blade. When the active control blade gives up the control, the standby automatically takes over and become the active control blade.

During normal operation, the CBC on the active control blade periodically scans the backplane. If a new blade is plugged in, it will be automatically powered up.

5       The active control blade register itself with NMS, and can take commands from NMS for controlling other blades in the chassis, such as checking their presence and status, power up/down or power cycle a blade, etc. The non-controlling blades also register themselves to NMS and can take commands from NMS to reboot or power down.

From the management point of view, each blade is a standalone computer.

10       Besides its application functionalities, each blade has management software to monitor the health of the application software, system load and performance, as well as hardware related parameters such as CPU temperature, fan speed, and power supply voltage. The blade management software functionality is shown in FIG. 10.

15       The streaming application threads 1002 put their health and load information into a shared memory area periodically. The management monitor thread 1004 scans the area to analyze the status of the threads and the system. In addition to periodically reporting the state information to NMS through a SNMP agent 1006, appropriate actions as known in the art are taken when an abnormal state is detected.

20       As previously described, a service token based authentication scheme is employed as the precursor for any data transfer requested by a subscriber's media console. FIG. 11 shows the access control schemes, where "sk" indicates a secret key. Secret keys are established only between a system component, such as the media console 206 or the media station 102, and Authentication Server 1102. All other accesses among the system components are controlled by Kerberos style tokens granted by the Authentication  
25       Server. This reduces the number of secret keys distributed among the components, and makes adding new components simpler. An mc\_token 1104 is passed by the media console to the media station to obtain streaming services. A cp\_token 1106 is passed by a media station for data transfer between media stations.

A media console possesses two numbers, MC\_ID and MC\_Key. Those numbers can be burned into a chip in the box, be on a Smartcard, or be on any form of non-volatile memory in the box. When a subscriber signs up for the service, the Subscriber Management system records the numbers and associates them with the user account.

5 MC\_ID and MC\_Key will be subsequently passed to the Authentication Server. FIG. 12 depicts the process of authentication.

A media console 206 when it powers up, after obtaining IP, sends an authentication request 1202 [which for the embodiment disclosed comprises MC\_ID, {MC\_ID, MC\_IP, Other info, salt, checksum}\_MC\_Key] to the Authentication Server  
10 1102. Note: { $x$ }\_ $k$  denotes that the message  $x$  is encrypted by  $k$ .

The Authentication Server finds the record of the media console using MC\_ID, decrypts the message, and generates a session key, MC\_SK, and an access\_token for the media console. For an exemplary embodiment access\_token = {MC\_SK, service code, timestamp, checksum}\_MS\_SK, where MS\_SK is a secret key established previously  
15 between the authentication server and the media station that serves the media console; “service code” indicates what services the token can be used for. The Authentication Server calculates the “seed key” for MC\_SK. The Authentication Server replies 1204 to the media console with [ {access\_token, MS\_IP, salt, checksum}\_MC\_Key ]. The MC decrypts the message with MC\_Key and obtains mc\_token and the IP address of the  
20 Media Director that it should contact. The mc\_token will be kept until the media console shuts down, or the Authentication Server sends a new one. The media console sends 1206 mc\_token to the application Server in the media station when requesting a media program, or the EPG server for browsing the EPG.

The implementation of the access tokens and encryption of the content provided  
25 over the system in an exemplary embodiment employs SecureMedia’s Encryptonite System for secure content delivery and access right control.

Having now described the invention in detail as required by the patent statutes, those skilled in the art will recognize modifications and substitutions to the specific

embodiments disclosed herein. Such modifications are within the scope and intent of the present invention as defined in the following claims.